EXPERIMENT – 7

Aim:- Write an application to simulate traffic lights and calculator using GridbagLayout.

7.1) Write an application to simulate traffic lights. The program lets the user select one of the three lights red, yellow and green. Upon selecting a menu item, the light is turned on and there is only one light on at a time

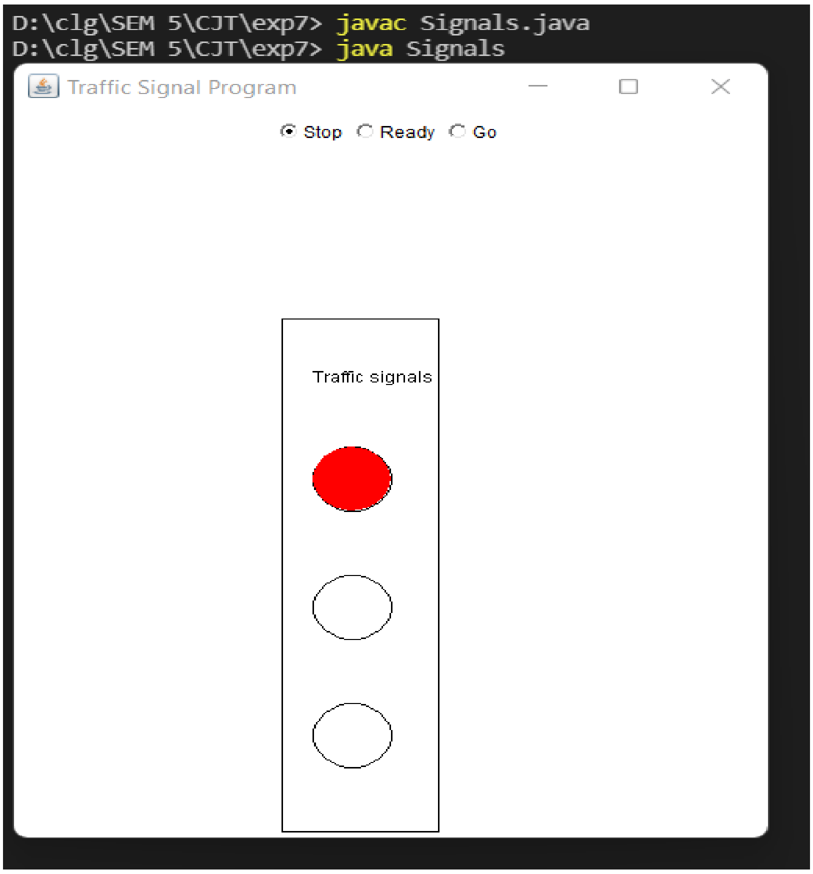
Code:-

```
import java.awt.*;
import java.awt.event.*;


public class Signals extends Frame implements ItemListener {


   String msg = "";


   Signals() {


      setTitle("Traffic Signal Program");
      setSize(500, 500);
      setVisible(true);


      Checkbox stop, ready, go;
      CheckboxGroup cbg = new CheckboxGroup();
      stop = new Checkbox("Stop", cbg, false);
      ready = new Checkbox("Ready", cbg, false);
      go = new Checkbox("Go", cbg, false);




      setLayout(new FlowLayout(FlowLayout.CENTER));
      add(stop);
      add(ready);
```

```
        add(go);
        stop.addItemListener(this);
        ready.addItemListener(this);


          go.addItemListener(this);

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });

    }



          public static void main(String[] args) {
        Frame f = new Signals();
    }

    public void itemStateChanged(ItemEvent ie) {
        msg = (ie.getItem()).toString();
        repaint();
    }



    public void paint(Graphics g) {

        g.drawString("Traffic signals", 200, 250);
        g.drawOval(200, 300, 50, 50);
        g.drawOval(200, 400, 50, 50);
        g.drawOval(200, 500, 50, 50);
        g.drawRect(180, 200, 100, 400);

        if (msg.equals("Stop"))
          {
            g.setColor(Color.red);

            g.fillOval(200, 300, 50, 50);
```

```
      } else if (msg.equals("Ready")) {
         g.setColor(Color.yellow);

         g.fillOval(200, 500, 50, 50);


      } else {
         g.setColor(Color.green);
         g.fillOval(200, 400, 50, 50);
      }
   }
}
```

Input/Output:



```
D:\clg\SEM 5\CJT\exp7> javac Signals.java
D:\clg\SEM 5\CJT\exp7> java Signals
```

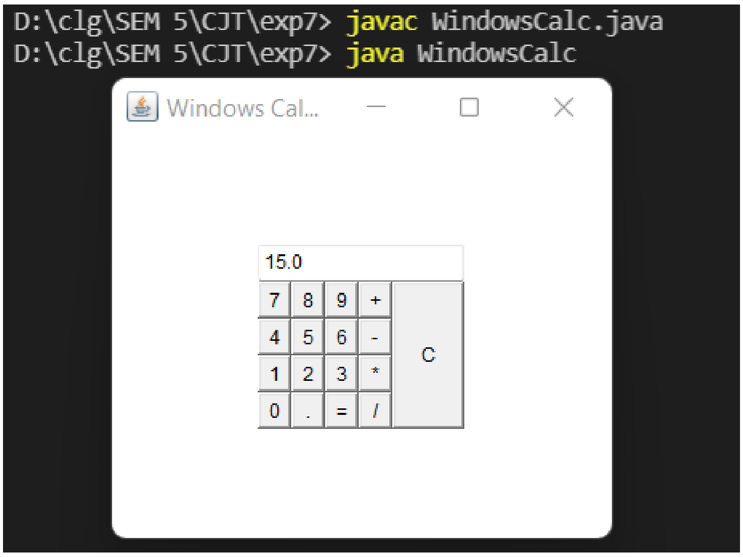
● Stop ○ Ready ○ Go

7.2) Using GridBagLayout to lay out, implement Windows' 98 calculator like application. Implement four basic arithmetic operations.

Code:

```
import java.awt.*;
import java.awt.event.*;

public class WindowsCalc extends Frame implements ActionListener {
    private TextField tf;
    private Button b0, b1, b2, b3, b4, b5, b6, b7, b8, b9, b10, b11, b12, b13, b14,
b15, b16;
    private GridBagLayout gbl;
    private GridBagConstraints gbcn;

    private int a;
    private Double n;
    private String s1, s2, s3, s4, s5;

    public static void main(String args[]) {
        Frame f = new WindowsCalc();
        f.setSize(300, 300);
        f.setVisible(true);
        f.setTitle("Windows Calculator");
    }

    public void addComp(Component c, GridBagLayout gbl, GridBagConstraints
 gbcn, int ro, int co, int nr, int nc, int wx,
            int wy) {
        gbcn.gridx = co;
        gbcn.gridy = ro;
        gbcn.gridwidth = nc;
        gbcn.gridheight = nr;
        gbcn.weightx = wx;
        gbcn.weighty = wy;

        gbl.setConstraints(c, gbcn);

        add(c);
    }

    public WindowsCalc() {
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
        tf = new TextField("", 15);
        b0 = new Button("0");
        b1 = new Button("1");
        b2 = new Button("2");
        b3 = new Button("3");
        b4 = new Button("4");
        b5 = new Button("5");
        b6 = new Button("6");
        b7 = new Button("7");
        b8 = new Button("8");
        b9 = new Button("9");
        b10 = new Button("+");
        b11 = new Button("-");
        b12 = new Button("*");
        b13 = new Button("/");
        b14 = new Button(".");
        b15 = new Button("=");
        b16 = new Button("C");

        gbl = new GridBagLayout();
        gbcn = new GridBagConstraints();
        setLayout(gbl);

        gbcn.fill = GridBagConstraints.BOTH;
        gbcn.anchor = GridBagConstraints.WEST;
        addComp(tf, gbl, gbcn, 0, 0, 1, 5, 0, 0);

        addComp(b16, gbl, gbcn, 1, 4, 4, 1, 0, 0);

        addComp(b7, gbl, gbcn, 1, 0, 1, 1, 0, 0);
        addComp(b8, gbl, gbcn, 1, 1, 1, 1, 0, 0);
        addComp(b9, gbl, gbcn, 1, 2, 1, 1, 0, 0);
        addComp(b10, gbl, gbcn, 1, 3, 1, 1, 0, 0);

        addComp(b4, gbl, gbcn, 2, 0, 1, 1, 0, 0);
        addComp(b5, gbl, gbcn, 2, 1, 1, 1, 0, 0);
        addComp(b6, gbl, gbcn, 2, 2, 1, 1, 0, 0);
        addComp(b11, gbl, gbcn, 2, 3, 1, 1, 0, 0);
        addComp(b1, gbl, gbcn, 3, 0, 1, 1, 0, 0);
        addComp(b2, gbl, gbcn, 3, 1, 1, 1, 0, 0);
        addComp(b3, gbl, gbcn, 3, 2, 1, 1, 0, 0);
        addComp(b12, gbl, gbcn, 3, 3, 1, 1, 0, 0);

        addComp(b0, gbl, gbcn, 4, 0, 1, 1, 0, 0);
        addComp(b14, gbl, gbcn, 4, 1, 1, 1, 0, 0);
        addComp(b15, gbl, gbcn, 4, 2, 1, 1, 0, 0);
        addComp(b13, gbl, gbcn, 4, 3, 1, 1, 0, 0);

        b0.addActionListener(this);
        b1.addActionListener(this);
        b2.addActionListener(this);
        b3.addActionListener(this);
        b4.addActionListener(this);
        b5.addActionListener(this);
        b6.addActionListener(this);
        b7.addActionListener(this);
        b8.addActionListener(this);
        b9.addActionListener(this);
        b10.addActionListener(this);
        b11.addActionListener(this);
        b12.addActionListener(this);
        b13.addActionListener(this);
        b14.addActionListener(this);
        b15.addActionListener(this);
        b16.addActionListener(this);

    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == b0) {
            s1 = tf.getText();
            s2 = "0";
            s3 = s1 + s2;
            tf.setText(s3);
        }
        if (e.getSource() == b1) {
            s1 = tf.getText();
            s2 = "1";
            s3 = s1 + s2;
            tf.setText(s3);
        }
        if (e.getSource() == b2) {
            s1 = tf.getText();
            s2 = "2";
            s3 = s1 + s2;
            tf.setText(s3);
        }
        if (e.getSource() == b3) {
            s1 = tf.getText();
            s2 = "3";
            s3 = s1 + s2;
            tf.setText(s3);
        }
        if (e.getSource() == b4) {
            s1 = tf.getText();
            s2 = "4";
            s3 = s1 + s2;
            tf.setText(s3);
        }
        if (e.getSource() == b5) {
            s1 = tf.getText();
            s2 = "5";
            s3 = s1 + s2;
            tf.setText(s3);
        }
        if (e.getSource() == b6) {
            s1 = tf.getText();
            s2 = "6";
            s3 = s1 + s2;
            tf.setText(s3);
        }
        if (e.getSource() == b7) {
            s1 = tf.getText();
            s2 = "7";
            s3 = s1 + s2;
            tf.setText(s3);
        }
        if (e.getSource() == b8) {
            s1 = tf.getText();
            s2 = "8";
            s3 = s1 + s2;
            tf.setText(s3);
        }
        if (e.getSource() == b9) {
            s1 = tf.getText();
            s2 = "9";
            s3 = s1 + s2;
            tf.setText(s3);
        }
        if (e.getSource() == b14) {
            s1 = tf.getText();
            s2 = ".";
            s3 = s1 + s2;
            tf.setText(s3);
        }
        if (e.getSource() == b10) {
            s4 = tf.getText();
            tf.setText("");
            a = 1;
        }
        if (e.getSource() == b11) {
            s4 = tf.getText();
            tf.setText("");
            a = 2;
        }
        if (e.getSource() == b12) {
            s4 = tf.getText();
            tf.setText("");
            a = 3;
        }
        if (e.getSource() == b13) {
            s4 = tf.getText();
            tf.setText("");
            a = 4;
        }

        if (e.getSource() == b15) {
            s5 = tf.getText();
            if (a == 1) {
                n = Double.parseDouble(s4) + Double.parseDouble(s5);
                tf.setText(String.valueOf(n));
            }
            if (a == 2) {
                n = Double.parseDouble(s4) - Double.parseDouble(s5);
                tf.setText(String.valueOf(n));
            }
            if (a == 3) {
                n = Double.parseDouble(s4) * Double.parseDouble(s5);
                tf.setText(String.valueOf(n));
            }
            if (a == 4) {
                if ((Double.parseDouble(s5)) == 0) {
                    tf.setText("Cannot Divide By Zero");
                } else {
                    n = Double.parseDouble(s4) / Double.parseDouble(s5);
                    tf.setText(String.valueOf(n));
                }
            }
        }
        if (e.getSource() == b16) {
            tf.setText("");
        }
    }
}
```

Input/Output:-



```
D:\clg\SEM 5\CJT\exp7> javac WindowsCalc.java
D:\clg\SEM 5\CJT\exp7> java WindowsCalc
```

Experiment-8

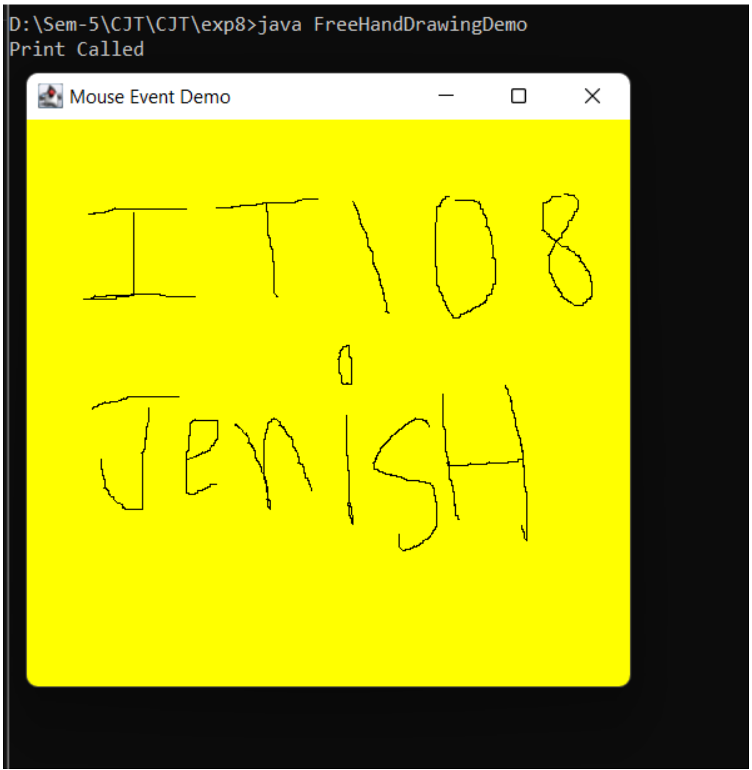
Aim:- Write a program that uses the concept of Applet and Exception Handling.

8.1) Write an Applet that allows free hand drawing. (Pencil tool of Paint Brush Application).

Code:-

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
/*
<applet code = "FreeHandDrawingDemo" height="400" width="400">
</applet>
*/
public class FreeHandDrawingDemo extends Applet {
    public static void main(String s[]) {
        Frame f = new Frame("Mouse Event Demo");
        FreeHandDrawingDemo fhd = new FreeHandDrawingDemo();
        fhd.init();
        f.add("Center",fhd);
        f.setSize(400,400);
        f.setVisible(true);
    }
    public void init() {
        Canvas c = new PaintCanvas();
        c.setBackground(Color.yellow);
        setLayout(new BorderLayout());
        add("Center",c);
    }
}
class PaintCanvas extends Canvas implements MouseMotionListener,MouseListener {
    final int circlesize = 20;
    private Point lineStart = new Point(0,0);
    public PaintCanvas() {
        addMouseMotionListener(this);
        addMouseListener(this);
    }
    public void mouseClicked(MouseEvent e) {}
    public void mouseReleased(MouseEvent e) {}
    public void mouseEntered(MouseEvent e) {}
    public void mouseExited(MouseEvent e) {}
    public void mousePressed(MouseEvent e) {
        lineStart.move(e.getX(),e.getY());
    }
    public void mouseDragged(MouseEvent e) {
        Graphics g = getGraphics();
        if(e.isMetaDown()) {
            g.setColor(getBackground());
            g.fillOval(e.getX() - (circlesize/2),e.getY() - (circlesize/2),circlesize,circlesize);
        } else {
            g.setColor(Color.black);
            g.drawLine(lineStart.x,lineStart.y,e.getX(),e.getY());
        }
        lineStart.move(e.getX(),e.getY());
    }
    public void mouseMoved(MouseEvent e) {
    }
    public void paint(Graphics g) {
        System.out.println("Print Called");
    }
}
```

Input/Output:-



```
D:\Sem-5\CJT\CJT\exp8>java FreeHandDrawingDemo
Print Called
```

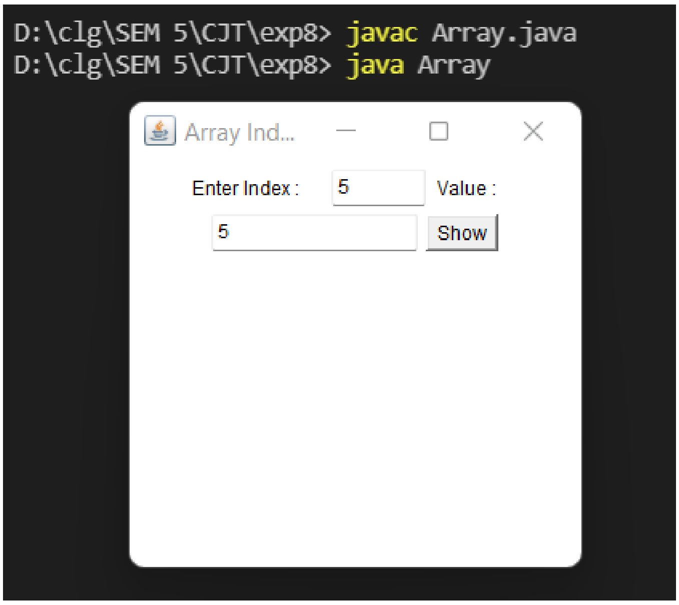
8.2) Write a program to meet following requirements:

- Create an array with 100 elements that are randomly chosen.
- Create a TextField to enter an array index and another TextField to display the array element at the specified index.
- Create a show Button to cause the array element to be displayed. If the specified index is out bound, display message “Out of Bound”.

Code:-

```
import java.awt.*;
import java.awt.event.*;

class Array extends Frame implements ActionListener {
    int a[] = new int[100];
    String s = "";
    TextField tf1, tf2;

    public static void main(String args[]) {
        Frame a = new Array();
        a.setTitle("Array Index Program");
        a.setSize(300, 300);
        a.setVisible(true);
    }

    Array() {
        for (int i = 0; i < 100; i++) {
            a[i] = (int) Math.random() + i;
        }
        Panel p1 = new Panel();
        Label l1 = new Label("Enter Index : ");
        Label l2 = new Label("Value : ");
        tf1 = new TextField("", 5);
        tf2 = new TextField("", 15);
        p1.add(l1);
        p1.add(tf1);
        p1.add(l2);
        p1.add(tf2);
        Button b1 = new Button("Show");
        p1.add(b1);
        b1.addActionListener(this);
        add(p1);
        setSize(500, 500);
        setVisible(true);
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getActionCommand().equals("Show")) {
            try {
                int i = Integer.parseInt(tf1.getText());
                s = "" + a[i];
                tf2.setText(s);
            } catch (Exception ae) {
                s = "Array Out Of Bound";
                tf2.setText(s);
            }
        }
    }
}
```

Input/Output:



```
D:\clg\SEM 5\CJT\exp8> javac Array.java
D:\clg\SEM 5\CJT\exp8> java Array
```

Enter Index : 5 Value :

5

Experiment-9

Aim:- Write the programs that uses the concept of Threads.

9.1) Write an application to display moving banner. Enter message to be moved, color and size through three different TextFields. Select direction (left or right) using Choice. On pressing start Button message should start moving.

Code:-

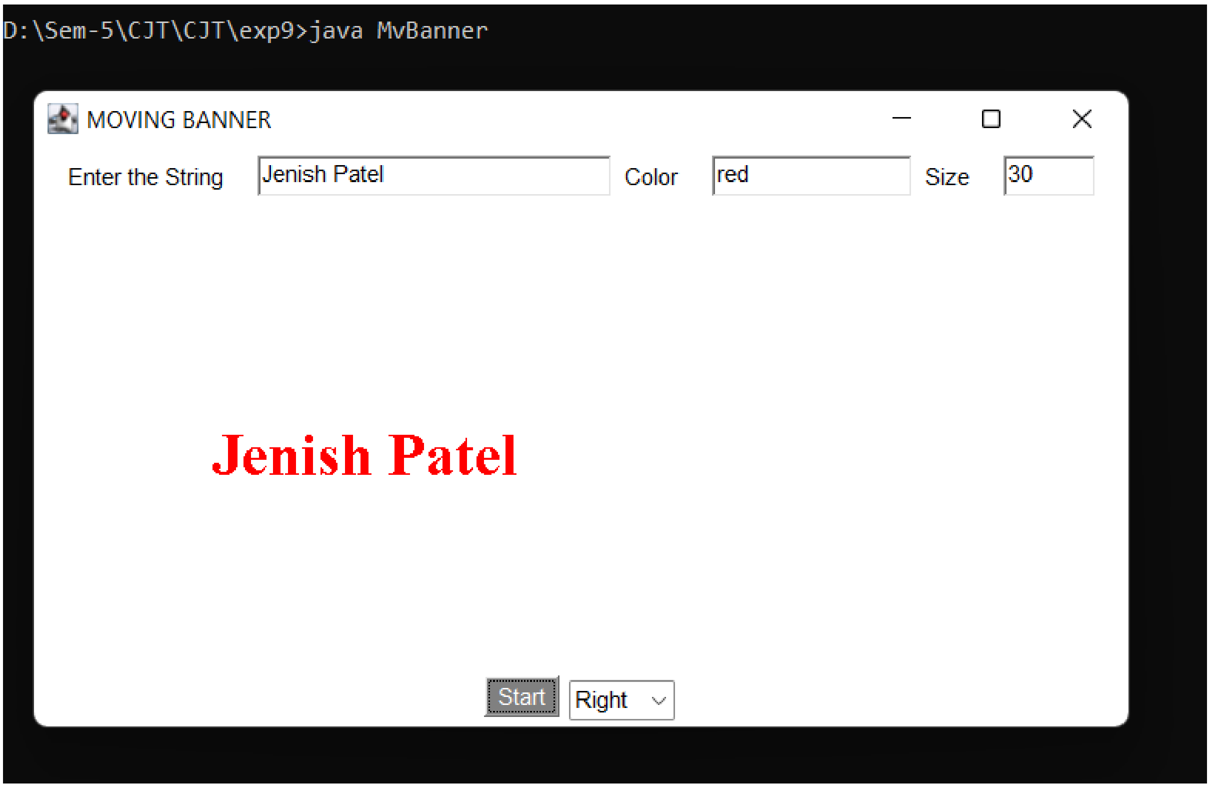
```
import java.awt.*;
import java.awt.event.*;
import java.util.*;

class MvBanner extends Frame implements Runnable,ActionListener
{

    Button b;
    TextField tf1;
    TextField tf2;
    TextField tf3;
    Label l1;
    Label l2;
    Label l3;
    Choice c;
    Thread t;
    String msg="";
    String col="";
    int x=200;
    int y=200;
    int z=0;

    public static void main(String[] args)
    {
        MvBanner f=new MvBanner("MOVING BANNER");
    }

    MvBanner(String title)
    {
        super(title);
        t=new Thread(this,title);
        //container containing components
```

```
     Panel p=new Panel();
     tf1=new TextField(20);
     tf2=new TextField(10);
     tf3=new TextField(3);
     l1=new Label("Enter the String");
     l2=new Label("Color");
     l3=new Label("Size");
     Panel p2=new Panel();
     b=new Button("Start");
     c=new Choice();
     c.add("Right");
     c.add("Left");

     setLayout(new BorderLayout());
     p.add(l1);
     p.add(tf1);
     p.add(l2);
     p.add(tf2);
     p.add(l3);
     p.add(tf3);
     add("North",p);
     p2.add(b);
     p2.add(c);
     add("South",p2);
     setSize(800,400);
     setVisible(true);

     //setBackground(Color.white);
     b.setBackground(Color.gray);
     b.setForeground(Color.white);
     b.addActionListener(this);

       //closing the window
     addWindowListener(new WindowAdapter()
     {
        public void windowClosing(WindowEvent e)
        {
           System.exit(0);
        }
     }
     );
  }
```

```
public void run()
{
    //Proper formatting of the movement of the banner.Handling exception
    try
    {
        while(true)
        {
            Thread.sleep(200);
            msg=tf1.getText();
            z=Integer.parseInt(tf3.getText());
            col=tf2.getText();

        if(c.getSelectedItem().equals("Left"))
            {
                x=x-20;
                if(x<(-100))
                x=800;
                repaint();
            }
            else
            {
                x=x+20;

             if(x>(800))
                x=0;
                repaint();
            }
        }
    }
    catch(InterruptedException e)
    {
        e.printStackTrace();
    }
}
public void paint(Graphics g)
{
    //Changing the colour of the text as per user requirement
    Font f=new Font("Serif",Font.BOLD,z);
    g.setFont(f);
    if(col.equals("red"))

    setForeground(Color.red);
```

```
        else if(col.equalsIgnoreCase("blue"))
        setForeground(Color.blue);

        else if(col.equalsIgnoreCase("orange"))
        setForeground(Color.orange);

        else if(col.equalsIgnoreCase("cyan"))
        setForeground(Color.cyan);

        else if(col.equalsIgnoreCase("magenta"))
        setForeground(Color.magenta);

        else if(col.equalsIgnoreCase("yellow"))
        setForeground(Color.yellow);

        else
        setForeground(Color.black);
        g.drawString(msg,x,y);
    }
    public void actionPerformed(ActionEvent ae)
    {
        //Start button is pressed - start moving the banner
        String s=ae.getActionCommand();

        if(ae.getSource() instanceof Button) //if source of event is button the do the
following
        {
            if(s.equals("Start"))
            t.start();
        }
    }
}
```

Input/Output:-



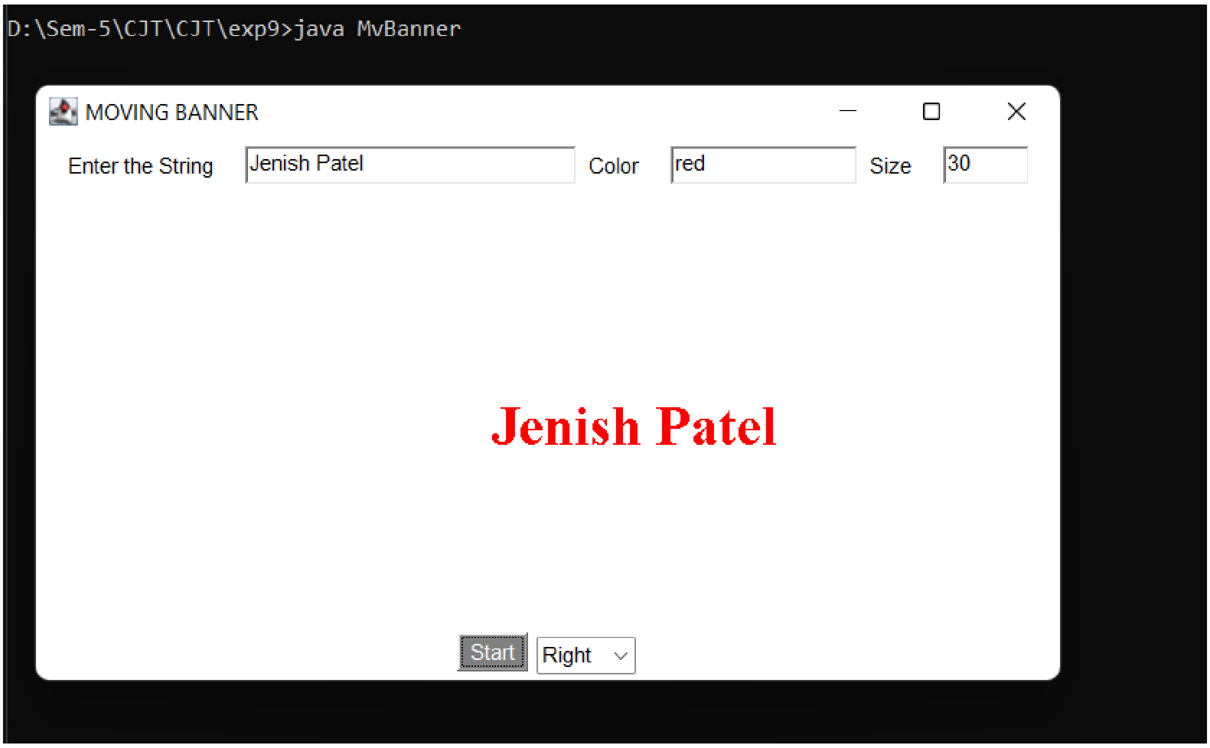
```
D:\Sem-5\CJT\CJT\exp9>java MvBanner
```

MOVING BANNER

Enter the String: Jenish Patel

Color: red

Size: 30



```
D:\Sem-5\CJT\CJT\exp9>java MvBanner
```

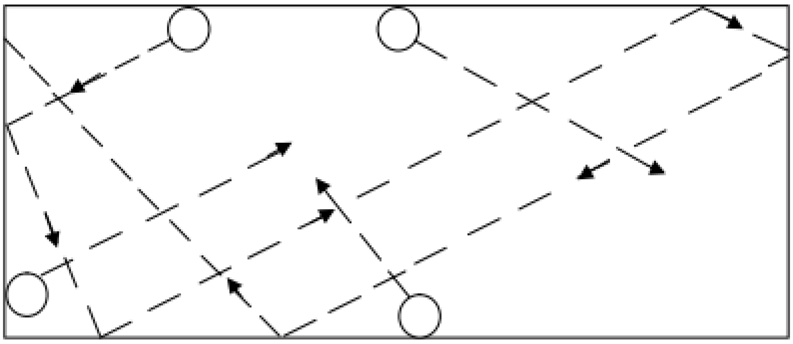
MOVING BANNER

Enter the String: Jenish Patel

Color: red

Size: 30

9.2) Write an applet that shows four moving objects (balls) as shown below:

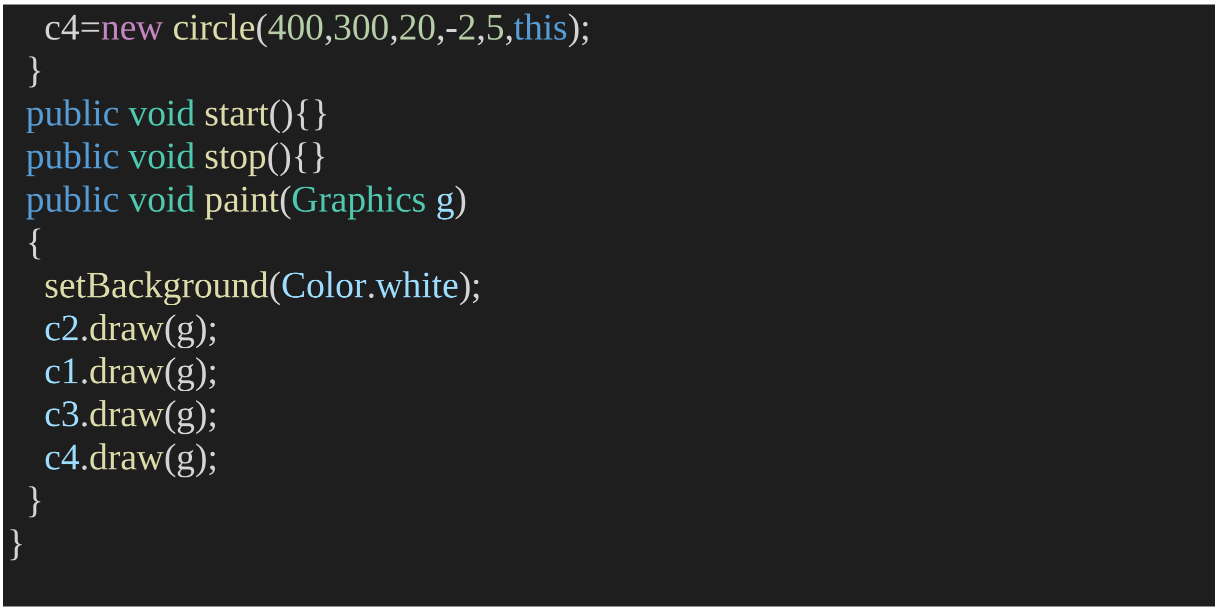


Four directions ( ) are used. On touching boundary, the object (ball) changes its direction and start moving in that changed direction.

Code:-

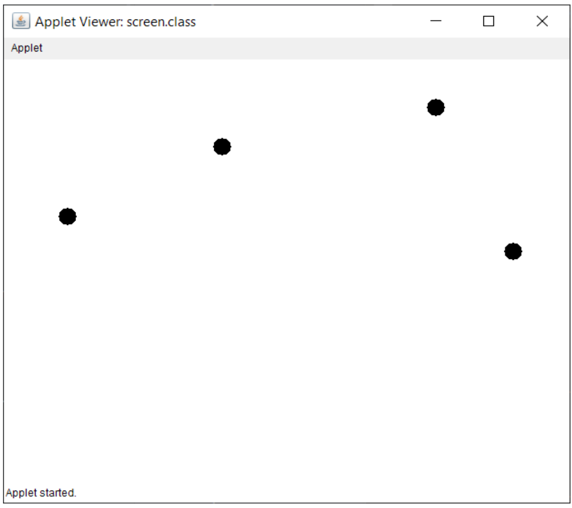
```
import java.awt.*;
import java.applet.*;
import java.awt.*;
/*
<applet code="screen.class" width="300" height="200">
</applet>
*/
class circle extends Thread
{
  public int x,y,r;
  public int inc;
  public int dir;
  public screen s1;
  public void draw(Graphics g)
  {
    g.fillOval(x,y,r,r);
  }
  public circle(int xx,int yy,int rr,int d,int l,screen s)
  {
    x=xx;y=yy;r=rr;
    dir=d;
    s1=s;
    inc=l;
    start();
  }
  public void run()
  {
    for(;;)
    {
     if(y>=300)
     {
      if(dir==1)
      {
       dir=2;
      }
      if(dir==-2)
      {
       dir=-1;
      }
     }
     if(y<=0)
     {
      if(dir==-1)
      {
       dir=-2;
      }
      if(dir==2)
      {
       dir=1;
      }
     }
     if(x>=600)
     {
      if(dir==1)
      {
       dir=2;
      }
      if(dir==2)
      {
       dir=-1;
      }
     }
     if(x<0)
     {
      if(dir==-1)
      {
       dir=2;
      }
      if(dir==-2)
      {
```

```
     dir=1;
    }
   }
   if(dir==1)
   {
    x=x+inc;
    y=y+inc;
   }
   else if(dir==2)
   {
    y=y-inc;
    x=x+inc;
   }
   else if(dir==-1)
   {
    x=x-inc;
    y=y-inc;
   }
   else if(dir==-2)
   {
    y=y+inc;
    x=x-inc;
   }
   s1.repaint();
   try
   {
    sleep(50);
   }
   catch(Exception e){
   }
  }
 }
}
public class screen extends Applet
{
 public circle c1,c2,c3,c4;
 public screen(){}
 public void init()
 {
  c1=new circle(0,300,20,1,10,this);
  c2=new circle(400,300,20,2,5,this);
  c3=new circle(400,0,20,-1,10,this);
```

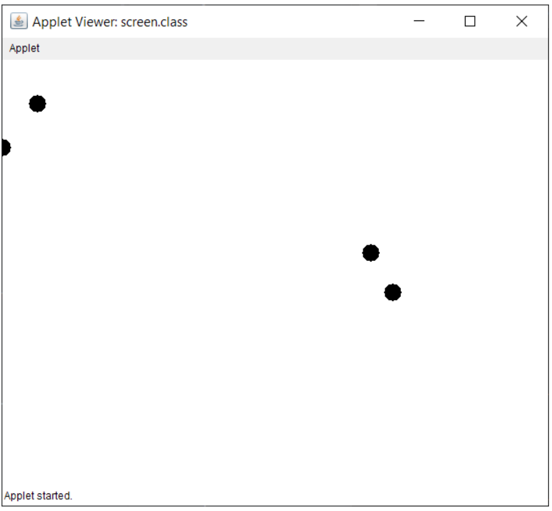


```
  c4=new circle(400,300,20,-2,5,this);
 }
 public void start(){}
 public void stop(){}
 public void paint(Graphics g)
 {
  setBackground(Color.white);
  c2.draw(g);
  c1.draw(g);
  c3.draw(g);
  c4.draw(g);
 }
}
```

Input/Output:



Applet started.



Applet started.

Experiment – 10

Aim:- Write a program that uses the concept of File I/O.

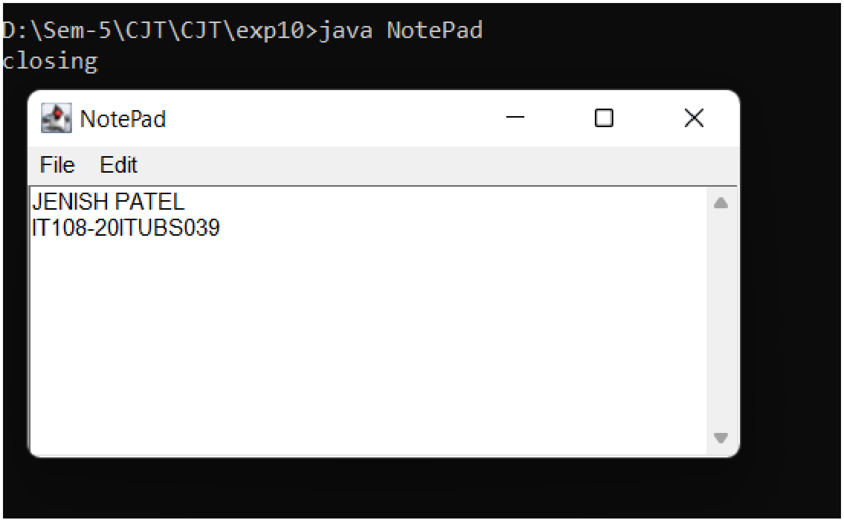
10.1) Write a program to implement Notepad like application.

Code:-

```
import java.awt.event.*;
import java.awt.*;
import java.io.*;

class NotePad extends Frame implements ActionListener,WindowListener
{
    int i=0;
    TextArea ta;
    static Frame p;
    BufferedReader br;
    FileInputStream fin;
    String path;
    NotePad()
    {
        setSize(800,600);
        setVisible(true);
        setTitle("NotePad");
        MenuBar m=new MenuBar();
        setMenuBar(m);

        Menu a=new Menu("File");
        m.add(a);
        MenuItem an=new MenuItem("New");
        an.addActionListener(this);
        a.add(an);
        MenuItem a1=new MenuItem("Open");
        a1.addActionListener(this);
        a.add(a1);
        MenuItem a2=new MenuItem("Save");
        a2.addActionListener(this);
        a.add(a2);
        MenuItem a3=new MenuItem("Save As");
        a3.addActionListener(this);
        a.add(a3);
        MenuItem ae=new MenuItem("Exit");
        ae.addActionListener(this);
        a.add(ae);

        Menu c=new Menu("Edit");
        m.add(c);
        MenuItem m21=new MenuItem("Cut");
        MenuItem m22=new MenuItem("Copy");
        MenuItem m23=new MenuItem("Paste");
        MenuItem m24=new MenuItem("Delete");
        c.add(m21);
        c.add(m22);
        c.add(m23);
        c.add(m24);
        m21.addActionListener(this);
        m22.addActionListener(this);
        m23.addActionListener(this);
        m24.addActionListener(this);


        ta=new TextArea();
        add(ta);
        addWindowListener(this);
    }
    public void actionPerformed(ActionEvent e)
    {
        if(e.getActionCommand().equals("New"))
        {
          NotePad m=new NotePad();
        }
        else if(e.getActionCommand().equals("Open"))
        {
            FileDialog fd=new FileDialog(p,"Open",FileDialog.LOAD);
            fd.setVisible(true);
            String s=fd.getFile();
            BufferedReader br=null;
            try
            {
              br=new BufferedReader(new FileReader(s));
              path=br.readLine();
```

```
                    boolean fl=true;
                    while(br!=null)
                    {
                        if(fl)
                        {
                            fl=false;
                            ta.append(path);
                        }
                        else
                        {
                            ta.append("\n"+path);
                        }
                        path=br.readLine();
                        if(path == null)
                            break;
                    }
                }catch(FileNotFoundException ef)
                {
                }
                catch(IOException eio)
                {
                }
        }
        else if(e.getActionCommand().equals("Save"))
        {

                FileDialog fd=new FileDialog(p,"Save",FileDialog.SAVE);
                fd.setVisible(true);
                System.out.println("file:-..."+fd.getFile());
                System.out.println("file:-..."+fd.getDirectory());

                try{
                    FileWriter fw=new
FileWriter(fd.getDirectory()+fd.getFile(),true);
                    fw.write(ta.getText());
                    fw.close();
                }catch(Exception ep)
                {System.out.println(ep);}
                System.out.println("Success...");
        }
        else if(e.getActionCommand().equals("Save As"))
        {
                FileDialog fd=new FileDialog(p,"Save",FileDialog.SAVE);
                fd.setVisible(true);
                System.out.println("file:-..."+fd.getFile());
                System.out.println("file:-..."+fd.getDirectory());
                try{
                    FileWriter fw=new FileWriter(fd.getDirectory()+fd.getFile());
                    fw.write(ta.getText());
                }catch(Exception ep)
                {System.out.println(ep);}
                System.out.println("Success...");
        }
     }
     public void windowActivated(WindowEvent arg0) {
        }
        public void windowClosed(WindowEvent arg0) {
        }
        public void windowClosing(WindowEvent arg0) {
           System.out.println("closing");
           dispose();
        }
        public void windowDeactivated(WindowEvent arg0){}
        public void windowDeiconified(WindowEvent arg0){}
        public void windowIconified(WindowEvent arg0){}
        public void windowOpened(WindowEvent arg0){}
     public static void main(String a[])
     {
        NotePad m=new NotePad();
     }
}
```

<u>Input/Output</u>:-



```
D:\Sem-5\CJT\CJT\exp10>java NotePad
closing
```

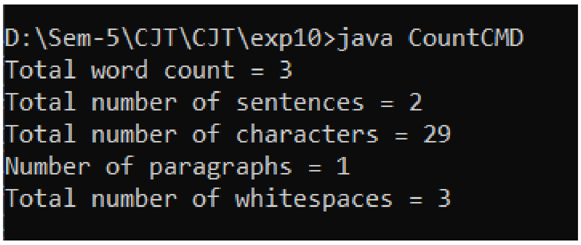
JENISH PATEL
IT108-20ITUBS039

10.2) Write a program that will count number of characters, words and lines in a file. The name should be passes as command line argument.

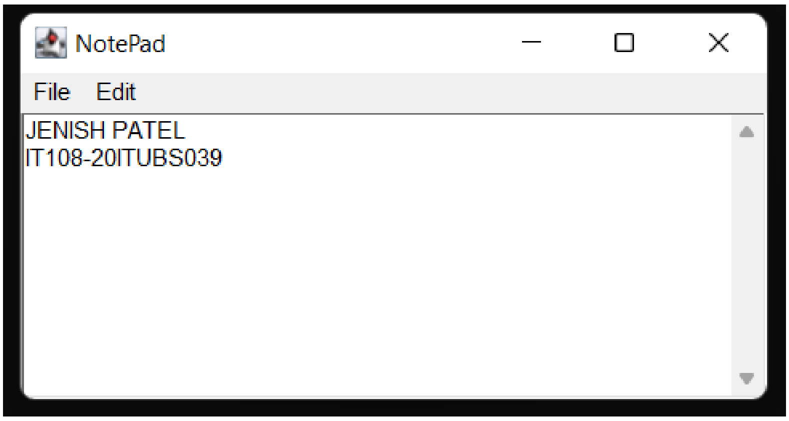
Code:-

```java
import java.io.*;

public class CountCMD
{
    public static void main(String[] args) throws IOException
    {
        File file = new  File("CMD.txt");
        FileInputStream fileStream = new FileInputStream(file);
        InputStreamReader input = new InputStreamReader(fileStream);
        BufferedReader reader = new BufferedReader(input);

        String line;

        // Initializing counters
        int countWord = 0;
        int sentenceCount = 0;
        int characterCount = 0;
        int paragraphCount = 1;
        int whitespaceCount = 0;

        // Reading line by line from the
        // file until a null is returned
        while((line = reader.readLine()) != null)
        {
            if(line.equals(""))
            {
                paragraphCount++;
            }
            if(!(line.equals("")))
            {

                characterCount += line.length();

                // \\s+ is the space delimiter in java
                String[] wordList = line.split("\\s+");

                countWord += wordList.length;
                whitespaceCount += countWord -1;
                // [!?.:]+ is the sentence delimiter in java
                String[] sentenceList = line.split("[!?.:]+");

                sentenceCount += sentenceList.length;
            }
        }

        System.out.println("Total word count = " + countWord);
        System.out.println("Total number of sentences = " + sentenceCount);
        System.out.println("Total number of characters = " + characterCount);
        System.out.println("Number of paragraphs = " + paragraphCount);
        System.out.println("Total number of whitespaces = " + whitespaceCount);
    }
}
```

Input/Output:



```
D:\Sem-5\CJT\CJT\exp10>java CountCMD
Total word count = 3
Total number of sentences = 2
Total number of characters = 29
Number of paragraphs = 1
Total number of whitespaces = 3
```



```
JENISH PATEL
IT108-20ITUBS039
```

Experiment – 11

Aim: **Write a program that uses the concept of socket programming.**

11.1) Implement multiple client chat application using socket programming.

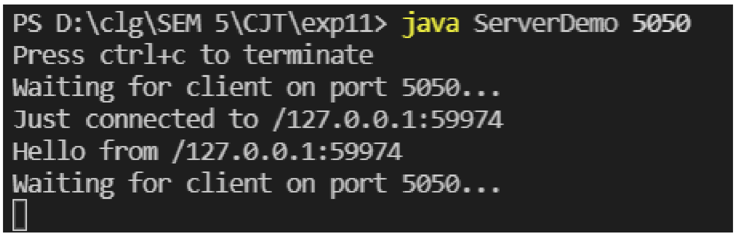
Server Code:-

```
import java.net.*;
import java.io.*;

public class ServerDemo extends Thread
{
  private ServerSocket ServerSocket;

  public ServerDemo(int port) throws IOException
  {
    ServerSocket = new ServerSocket(port);
    ServerSocket.setSoTimeout(100000);
  }

  public void run()
  {
    while(true)
    {
      try
      {
        System.out.println("Waiting for client on port " +
        ServerSocket.getLocalPort() + "...");
        Socket ServerDemo = ServerSocket.accept();
        System.out.println("Just connected to "
            + ServerDemo.getRemoteSocketAddress());
        DataInputStream in =
            new DataInputStream(ServerDemo.getInputStream());
        System.out.println(in.readUTF());
        DataOutputStream out =
            new DataOutputStream(ServerDemo.getOutputStream());
        out.writeUTF("Thank you for connecting to "
          + ServerDemo.getLocalSocketAddress() + "\nGoodbye!");
        ServerDemo.close();
      }
       catch(SocketTimeoutException s)
      {
        System.out.println("Socket timed out!");
        break;
      }catch(IOException e)
      {
        e.printStackTrace();
        break;
      }
    }
  }
  public static void main(String [] args)
  {
   System.out.println("Press ctrl+c to terminate");
    int port = Integer.parseInt(args[0]);
    try
    {
      Thread t = new ServerDemo(port);
      t.start();
    }catch(IOException e)
    {
      e.printStackTrace();
    }
  }
}
```

<u>Client Code</u>:-

```
import java.net.*;
import java.io.*;

public class ClientDemo
{
  public static void main(String [] args)
  {
    String serverName = args[0];
    int port = Integer.parseInt(args[1]);
    try
    {
      System.out.println("Connecting to " + serverName +
      " on port " + port);
      Socket ClientDemo = new Socket(serverName, port);
      System.out.println("Just connected to "
      + ClientDemo.getRemoteSocketAddress());
      OutputStream outToServer = ClientDemo.getOutputStream();
      DataOutputStream out = new DataOutputStream(outToServer);
      out.writeUTF("Hello from "
               + ClientDemo.getLocalSocketAddress());
      InputStream inFromServer = ClientDemo.getInputStream();
      DataInputStream in =
                new DataInputStream(inFromServer);
      System.out.println("Server says " + in.readUTF());
      ClientDemo.close();
    }catch(IOException e)
    {
      e.printStackTrace();
    }
  }
}
```

Input/Output:-



```
PS D:\clg\SEM 5\CJT\exp11> java ServerDemo 5050
Press ctrl+c to terminate
Waiting for client on port 5050...
Just connected to /127.0.0.1:59974
Hello from /127.0.0.1:59974
Waiting for client on port 5050...
```